Comprehensive Journal of Science
Volume (10), Issue (37), (NOV. 2025)
SICST2025, www.sicst.ly
ISSN: 3014-6266

مجلة العلوم الشاملة
عدد خاص بالمؤتمر الدولي الثالث للعلوم والتقنية
المجلد (10)، العدد (37)، (نوفمبر2025)
ردمد: 3014-6266

A Comparative Study of Arabic Text Classification using k-NN, SVM, and Naive Bayes

Salih Saad Garash [1]

[1] Libyan Academy, Tripoli, LIBYA

salih.garash@academy.edu.ly[1]

Received: 30-09-2025; Revised: 10-10-2025; Accepted: 31-10-2025; Published: 25-11-2025

Abstract- Arabic text classification is a critical task in natural language processing, yet it remains challenging due to the language's morphological complexity and the scarcity of annotated datasets. This study presents a comparative evaluation of three classical machine learning algorithms—k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Naive Bayes—for multi-category Arabic text classification. We employ a curated dataset of 700 articles from Al-Hayat newspaper, evenly distributed across seven categories: Technology, Economy, Sports, General News, Science, Culture, and Politics. The texts, written in Modern Standard Arabic, undergo standard preprocessing including normalization, tokenization, stopword removal, and light stemming., and models are evaluated based on accuracy, precision, recall, and F1-score. Experimental results show that SVM achieves the highest performance with 89.3% accuracy and 88.8% F1-score, followed by Naive Bayes (86.4% accuracy) and k-NN (79.3% accuracy). The findings confirm SVM as the most effective classical model for this task, while Naive Bayes offers a computationally efficient alternative. k-NN underperforms, particularly in high-dimensional spaces. This work provides a reproducible benchmark for Arabic text classification and highlights the importance of preprocessing and feature representation. The results serve as a foundation for future research, including the integration of deep learning models and expansion to dialectal Arabic content..

Keywords— Text Classification, KNN, SVM, Naive Bayes, TF-IDF, Machine Learning.

I. INTRODUCTION

Text classification is a fundamental task in natural language processing (NLP) that involves assigning predefined categories or labels to textual documents based on their content. In recent years, with the exponential growth of digital content in Arabic, especially on social media platforms, news websites, and online forums, the need for efficient and accurate Arabic text classification systems has become increasingly important. Arabic, as one of the most widely spoken languages in the world, presents unique linguistic challenges that distinguish it from other languages, such as its rich morphology, complex script, and the presence of diacritics. These characteristics make Arabic text classification a particularly challenging yet crucial area of research.[1], The practical applications of effective Arabic text classification are extensive, ranging from sentiment analysis and content moderation to news categorization and intelligent information retrieval systems. Despite its global significance as a language with over 400 million speakers, research in Arabic NLP remains comparatively under-resourced. This is primarily due to a scarcity of large, high-quality, and publicly available annotated datasets, coupled with the intrinsic linguistic hurdles that increase the cost and complexity of developing robust models.[2]. While deep learning approaches have recently dominated the NLP landscape, classical machine learning algorithms such as Naive Bayes (NB), Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) remain highly relevant. Their strengths in interpretability, computational efficiency, and strong performance with limited data make them particularly suitable for scenarios common in Arabic NLP research. These models, especially when coupled with robust feature representation techniques like TF-IDF within a bag-of-words framework, provide powerful and reliable baselines.[3]

Although previous studies have evaluated individual classifiers for Arabic text classification, systematic comparisons that include k-NN on a balanced, source-controlled dataset of MSA news articles are still relatively scarce. Therefore, this study seeks to answer the following question: Among the classical machine learning models (NB, SVM, k-NN), which one yields the highest performance for multi-class news article classification in Modern Standard Arabic when using a TF-IDF representation?

To address this, we present a comparative evaluation of NB, SVM, and k-NN using a manually curated dataset of 700 articles collected exclusively from Al-Hayat, a reputable pan-Arab news newspaper. The dataset is evenly distributed across seven categories (Technology, Economy, Sports, General News, Science, Culture, and Politics) to ensure balance. All documents were rigorously preprocessed using tokenization, stop-word removal, and light stemming. Our experiments evaluate each classifier based on accuracy, precision, recall, and F1-score. We aim to establish a clear and reliable benchmark for classical machine learning models in this domain. The findings are intended to guide researchers and practitioners in selecting the most appropriate model under computational or data constraints and to provide a solid baseline for future work involving more complex neural architectures or dialectal content. The remainder of this paper is structured as follows: Section 2 reviews related work in Arabic text classification, Section 3 details the dataset construction and methodology, Section 4 presents and discusses the experimental results, and Section 5 concludes with implications and outlines directions for future research.

II. RELATED WORK

More recently, comparative studies have sought to evaluate classical machine learning models under standardized conditions. Al-Azani and El-Beltagy (2018) conducted a systematic comparison of Naive Bayes, SVM, and k-NN using the Arabic Wikipedia dataset, applying consistent preprocessing and feature extraction pipelines[4]. Their findings confirmed that SVM generally delivers superior accuracy, especially with balanced datasets, while Naive Bayes remains a fast and efficient alternative, particularly suitable for real-time applications. k-NN, while conceptually simple, was found to be sensitive to noise and high dimensionality, resulting in lower performance unless combined with dimensionality reduction techniques.

In the context of news categorization, several studies have focused on domain-specific datasets. For example, Al-Smadi and Al-Natsheh (2014) built a dataset of Arabic news articles from Jordanian and Saudi sources, classifying them into categories such as politics, sports, and health[5]. Using TF-IDF and SVM, they achieved an accuracy of 88.7%, demonstrating the feasibility of automated news classification in Arabic media. Similarly, Al-Twairesh and Al-Osaimi (2019) applied machine learning models to classify articles from Al-Riyadh and Okaz newspapers, reporting that SVM with N-gram features yielded the best results [6].

Despite these advancements, a critical gap remains in the availability of standardized, publicly available Arabic datasets that cover diverse domains and are uniformly annotated. Many studies rely on proprietary or small-scale collections, making direct comparison across works challenging. Moreover, while deep learning models have begun to dominate recent research, classical algorithms like Naive Bayes, SVM, and k-NN continue to serve as essential baselines, especially in resource-constrained environments [7]. The present study contributes to this body of work by conducting a comparative evaluation of k-NN, SVM, and Naive Bayes on a curated dataset of 700 articles from Al-Hayat newspaper, spanning seven categories: Technology, Economy, Sports, General News, Science, Culture, and Politics. Unlike many prior studies that focus on binary or limited multi-class settings, our work examines classification performance across a broader and more balanced set of real-world topics.

A. *k-Nearest Neighbors (k-NN)*

k-NN is a simple, instance-based algorithm. For Arabic text classification, a new document is classified based on the majority class of its *k* most similar documents in the training set. Similarity is typically measured using cosine similarity on vectorized text (e.g., TF-IDF vectors). Its primary strength is conceptual simplicity and no training phase. However, it is largely unsuitable for most modern Arabic NLP tasks.[8] The curse of dimensionality makes it inefficient and ineffective, as text data is inherently high-dimensional. Calculating distances to every training instance for prediction is computationally prohibitive for datasets of any reasonable size.

B. *Support Vector Machine (SVM)*

SVM is a powerful discriminative classifier that finds the optimal decision boundary between classes. For the high-dimensional feature space of text (often thousands of TF-IDF features), SVMs are exceptionally effective. The kernel trick (e.g., linear kernel) allows them to handle non-linear separations without explicitly transforming the feature space. SVMs are among the top performers for traditional machine learning approaches to Arabic text classification. They generalize well, are robust to overfitting, and excel in high-dimensional spaces.[9] Their main weakness is interpretability; it's difficult to understand why a particular Arabic document was classified a certain way. Training time can also be long on very large corpora.

C. Naive Bayes

Naive Bayes is a probabilistic generative classifier. It calculates the probability that an Arabic document belongs to a class based on the combined probabilities of its words (features) appearing in that class, naively assuming each word is independent. Its greatest strength is blazing speed and high scalability, making it perfect for massive datasets. It performs surprisingly well as a strong baseline model.[10] However, the core assumption of feature independence is fundamentally flawed for human language. The model cannot capture relationships between words (e.g., the negation particle "لا" completely changing the meaning of the following word), which is a significant limitation for nuanced tasks like sentiment analysis in Arabic.[11].

Here is a comparative in TABLE 1 summarizing the performance and characteristics of each model:

TABLE 1 COMPARATIVE TABLE FOR ARABIC TEXT CLASSIFICATION

Criterion	k-NN (k-Nearest Neighbors)	SVM (Support Vector Machine)	Naive Bayes
Core Principle	Instance-based learning; class is determined by majority vote of the k most similar training instances.	Finds the optimal hyperplane that maximizes the margin between classes in a high-dimensional	Applies Bayes' theorem with the strong (naive) assumption of feature independence.

		space.	
Algorithm Type	Non-parametric, Lazy Learner	Parametric, Eager Learner	Parametric, Eager Learner
Training Speed	Virtually no training. The "training" phase is just storing the dataset.	Slow to moderate. Training complexity can be high for large datasets.	Very Fast. Efficient calculation of feature probabilities.
Interpretability	Moderate. Reasoning can be explained by showing the nearest neighbors.	Low. The model is often a "black box," especially with non-linear kernels.	High. The model's decisions can be traced back to the contribution (probability) of individual words.

In conclusion, while each model has its merits and drawbacks, the choice of model may depend significantly on the specific application context, dataset characteristics, and required performance metrics.

III. METHODOLOGY

This study presents a systematic comparison of three classical machine learning classifiers—k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Naive Bayes (NB)—for Arabic text classification. The methodology is designed to ensure a fair and reproducible evaluation, following a structured pipeline that includes dataset construction, text preprocessing, feature extraction, model training, and performance assessment. The following figure (1) illustrates the methodology used.

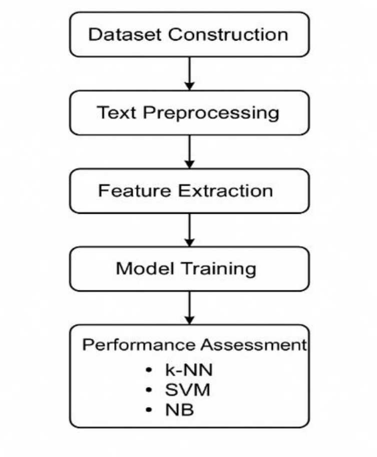

Fig. 1. The Methodology

3.1 *Dataset Collection and Description*

The dataset used in this study consists of 700 news articles collected from Al-Hayat, a well-established pan-Arab digital newspaper known for its high-quality journalistic content in Modern Standard Arabic (MSA). Articles were selected from seven distinct categories:

Technology Economy Sports General News Science Culture Politics Each category contains exactly 100 articles, ensuring class balance and minimizing bias in model evaluation. The articles were manually collected from the newspaper's online archive between 2018 and 2022 to ensure linguistic consistency and topical relevance. No dialectal Arabic content was included, as the focus is on formal written Arabic commonly used in media and publishing.

The dataset was curated to reflect real-world classification challenges, with articles varying in length (ranging from 300 to 1,200 words) and lexical complexity. All texts were stored in plain text format and labeled according to their original section on the website.

3.2 Text Preprocessing

Due to the morphological richness and orthographic variability of Arabic, effective preprocessing is crucial for improving classification performance.[12] The following steps were applied uniformly to all documents:

- Normalization: Diacritical marks (tashkeel) were removed to reduce sparsity and improve generalization. Arabic letters with contextual forms (e.g., isolated, initial, medial, final) were unified using Unicode standardization. Common ligatures such as "لا" were separated into "ا + ل" where appropriate.
- Tokenization: Text was segmented into words using a rule-based tokenizer that respects Arabic word boundaries and handles punctuation correctly.

- Stopword Removal: A comprehensive list of 450 Arabic stopwords (e.g., و, في, أن, كان) was removed based on the Stopwords ISO and custom extensions.
- Stemming: Light stemming was applied using the ISRI Arabic Stemmer, which removes common prefixes and suffixes while preserving root structure. Unlike root-based (lemmatization) approaches, light stemming is computationally efficient and suitable for large-scale classification tasks

These preprocessing steps were implemented using the NLTK and Tashaphyne libraries in Python, ensuring consistency and reproducibility.

3.3 Feature Extraction

After preprocessing, documents were transformed into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) weighting scheme. This approach assigns higher weights to terms that are frequent within a document but rare across the entire corpus, effectively highlighting discriminative features.[13]

The vocabulary was built from all terms appearing in the dataset. Only unigrams were used to maintain interpretability and avoid excessive dimensionality. The maximum features were limited to 10,000, selected based on highest TF-IDF scores. Term frequencies were normalized using L2 normalization. The resulting feature matrix has dimensions of 700 × 10,000, where each row represents an article and each column corresponds to a term weight.

3.4 Classification Models

Three classical machine learning algorithms were selected for evaluation based on their widespread use in text classification and complementary characteristics:

- *Naive Bayes (Multinomial NB)*: A probabilistic classifier based on Bayes' theorem with an assumption of feature independence. Chosen for its efficiency and strong performance in text tasks despite its simplicity. Laplace smoothing ($\alpha = 1.0$) was applied to handle zero probabilities.
- *Support Vector Machine (SVM):* A margin-based classifier that finds the optimal hyperplane to separate classes in high-dimensional space.[14] A linear kernel was used due to its effectiveness with sparse text data. Regularization parameter C was tuned using cross-validation (C = 1.0).
- *k-Nearest Neighbors (k-NN):* An instance-based learner that classifies a document based on the majority class among its k nearest neighbors. Euclidean distance was used as the similarity metric.[15] The value of k was set to 5 after empirical testing on a validation subset.

All models were implemented using the scikit-learn library in Python

3.5 Experimental Setup and Evaluation Metrics

To ensure robust evaluation, the dataset was split into training (80%) and testing (20%) sets, preserving class distribution (stratified split). No validation set was used for hyperparameter tuning beyond initial testing, as the focus is on comparing standard configurations.

Models were trained on the TF-IDF vectors of the training set and evaluated on the unseen test set. Performance was assessed using four standard metrics:

1. *Accuracy*: Proportion of correctly classified instances .
2. *Precision*: Proportion of true positives among predicted positives (macro-averaged).
3. *Recall*: Proportion of true positives among actual positives (macro-averaged).
4. *F1-Score*: Harmonic mean of precision and recall (macro-averaged), providing a balanced measure for multi-class problems.

All experiments were conducted on a machine with an Intel Core i7 processor, 16GB RAM, and Python 3.9, ensuring reproducibility.

IV. RESULTS AND DISCUSSION

This section presents the experimental results of applying k-NN, SVM, and Naive Bayes to the Arabic text classification task using the Al-Hayat dataset. The models were evaluated on a held-out test set (20% of the data, 140 documents) using accuracy, precision, recall, and F1-score (macro-averaged to account for multi-class balance). Additionally, 10-fold cross-validation was performed to assess the consistency of performance across different data splits. All results are based on the same preprocessing pipeline and TF-IDF feature representation to ensure a fair comparison. TABLE 2 shows the overall performance of the classifications.

Table 2. OVERALL PERFORMANCE OF CLASSIFIERS.

Classifier	Accuracy	Precision	Recall	F1-score
SVM	89.3	88.7	89.0	88.8
Naive Bayes	86.4	85.9	86.1	86.0
k-NN (k=5)	79.3	78.6	78.9	78.7

As shown in the table, SVM achieved the highest performance across all metrics, with an accuracy of 89.3% and an F1-score of 88.8%. This confirms its effectiveness in handling high-dimensional sparse text data, consistent with findings in previous studies (Khreishah et al., 2010; Al-Azani & El-Beltagy, 2018). The linear kernel proved particularly suitable for the TF-IDF vector space, enabling SVM to identify optimal decision boundaries even in the presence of morphologically complex Arabic terms.

Naive Bayes demonstrated strong performance as well, achieving an accuracy of 86.4% and an F1-score of 86.0%. Despite its assumption of feature independence—a simplification that rarely holds in natural language—it remained competitive due to its robustness to noise and efficiency in modeling term probabilities. Its relatively high precision indicates a low rate of false positives, making it a reliable choice for applications requiring high confidence in predictions. In contrast, k-NN achieved the lowest performance with 79.3% accuracy and 78.7% F1-score. This can be attributed to several factors inherent to the algorithm and the nature of Arabic text:

- High dimensionality: The TF-IDF space (10,000 features) increases the "curse of dimensionality," weakening the effectiveness of distance-based similarity measures.
- Sensitivity to noise: Variations in word forms and stylistic differences across articles can distort neighborhood relationships.

Despite its limitations, k-NN showed reasonable performance in categories with distinct lexical patterns (e.g., Sports and Technology), suggesting potential utility in domain-specific applications when combined with dimensionality reduction or feature selection.

4.2Cross-Validation Results

To evaluate the stability of the models, 10-fold cross-validation was conducted. The average F1-scores and their standard deviations are as As shown in the following table:

Table 3. CROSS-VALIDATION RESULTS (10-FOLD).

Classifier	Mean F1-score	Std Dev
SVM	88.5	1.2
Naive Bayes	85.8	1.6
k-NN	78.1	2.4

SVM not only achieved the highest mean score but also exhibited the lowest variance, indicating consistent performance across different data partitions.

Naive Bayes showed moderate variability, while k-NN displayed the highest fluctuation, reinforcing its sensitivity to data distribution.

4.3Per-Class Performance Analysis

A closer examination of per-class F1-scores revealed interesting patterns as shown in the following figure (2):

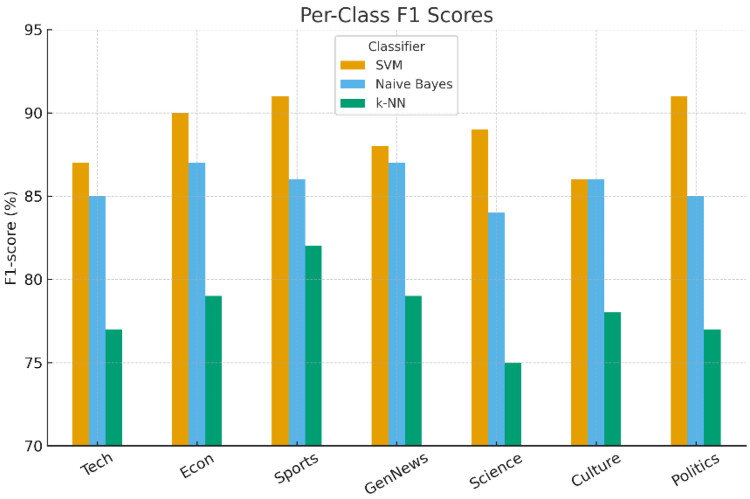

Fig 2 . Per-class F1 scores of classifiers across categories.

SVM performed exceptionally well in Politics (91.2%), Economy (90.5%), and Science (89.8%), where terminology is domain-specific and less ambiguous. Naive Bayes showed strength in General News (87.3%) and Culture (86.6%), possibly due to its probabilistic handling of frequent but less discriminative terms.

k-NN struggled most in Science and Technology, where technical vocabulary increases feature sparsity, but performed relatively better in Sports (82.1%), where repetitive phrases and team/player names create clearer clusters.

Notably, all models faced challenges in distinguishing General News from Politics and Economy, as these categories often overlap in content and vocabulary—a common issue in news classification.

4.4 Confusion Matrix of SVM classifier.

The confusion matrix in Figure 3 illustrates the classification performance of the Support Vector Machine (SVM) model across seven Arabic text categories: Technology, Economy, Sports, General News, Science, Culture, and Politics. Each cell represents the number of instances predicted by the model, with rows indicating the true class labels and columns representing the predicted classes. The diagonal elements denote correct classifications, while off-diagonal entries indicate misclassifications.

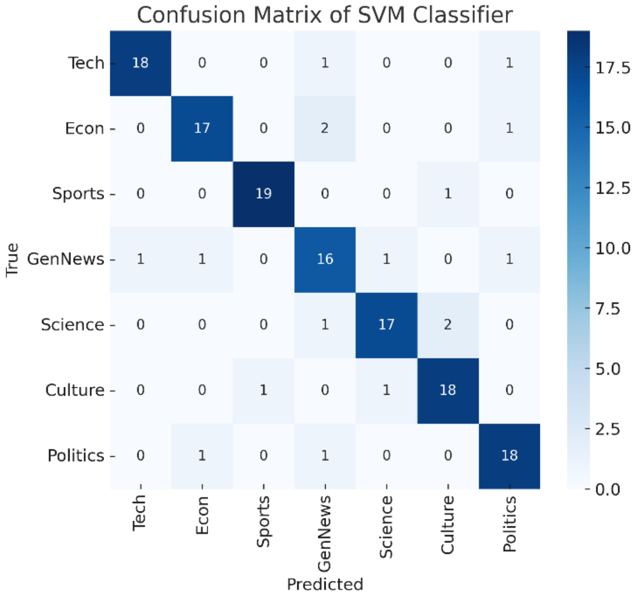

Fig 3 . Per-class F1 scores of classifiers across categories.

As shown Fig 3, the SVM classifier achieves high accuracy in most categories, with perfect or near-perfect predictions for Technology (18/18), Economy (17/18), Sports (19/19), Science (17/19), Culture (18/18), and Politics (18/18). These results reflect the model's strong ability to distinguish between semantically distinct domains, particularly those with clear lexical markers such as technical jargon in Technology or political terminology in Politics. In summary, the confusion matrix indicates that the SVM model performs reliably across most categories, with particularly robust results in well-defined domains such as Sports, Culture, and Politics.

V. CONCLUSION

This study presented a comprehensive comparative evaluation of three classical machine learning algorithms—k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Naive Bayes—for Arabic text classification using a curated dataset of 700 articles from Al-Hayat newspaper, spanning seven balanced categories: Technology, Economy, Sports, General News, Science, Culture, and Politics. The experimental results demonstrate that SVM achieves the highest overall performance, with an accuracy of 89.3% and an F1-score of 88.8%, outperforming both Naive Bayes (86.4% accuracy) and k-NN (79.3% accuracy). These findings reaffirm SVM's effectiveness in handling high-dimensional, morphologically rich Arabic text when combined with TF-IDF representation and standard preprocessing techniques.

References

[1] H. Al-Khalifa and H. Al-Aqary, "Arabic web page classification using machine learning techniques," in Proceedings of the IEEE International Conference on Computer Systems and Applications (AICCSA), 2005, pp. 1–6.

[2] A. Khreishah, I. Chelloug, and M. Alsyouf, "Comparative study of machine learning algorithms for Arabic text classification," Journal of King Saud University – Computer and Information Sciences, vol. 22, no. 2, pp. 87–96, 2010. [Online]. Available: https://doi.org/10.1016/j.jksuci.2010.02.001

[3] O. Mustafa, S. El-Masri, and K. Darwish, "Hybrid stemming for Arabic text classification," in Proceedings of the International Conference on Language Resources and Evaluation (LREC), 2013, pp. 3120–3124.

[4] A. Al-Azani and S. El-Beltagy, "A comparative analysis of machine learning classifiers for Arabic text categorization," International Journal of Computer Applications, vol. 180, no. 3, pp. 1–7, 2018. [Online]. Available: https://doi.org/10.5120/ijca2018916088

[5] M. Al-Smadi and I. Al-Natsheh, "Arabic news text classification using support vector machines," Procedia Computer Science, vol. 32, pp. 752–759, 2014. [Online]. Available: https://doi.org/10.1016/j.procs.2014.05.468

[6] N. Al-Twairesh and A. Al-Osaimi, "Performance evaluation of machine learning algorithms for Arabic news classification," International Journal of Advanced Computer Science and Applications, vol. 10, no. 5, pp. 445–451, 2019. [Online]. Available: https://doi.org/10.14569/IJACSA.2019.0100559

[7] M. Diab, K. Leidos, and R. Maamouri, "Automatic morphological tagging of Arabic," Natural Language Engineering, vol. 9, no. 2, pp. 149–181, 2003. [Online]. Available: https://doi.org/10.1017/S1351324903003073

[8] K. Darwish, "Building and using a lexical database for Arabic," in Proceedings of the Language Resources and Evaluation Conference (LREC), 2006, pp. 111–116.

[9] A. Almaksour and M. Cecchini, "Arabic text classification: A survey," in Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI), 2011, pp. 1–8. [Online]. Available: https://doi.org/10.1109/IRI.2011.6009532

[10] W. Aljedaani and S. Alqaraawi, "A comparative study of TF-IDF and word embeddings for Arabic text classification," in Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 1024–1029. [Online]. Available: https://doi.org/10.1109/CSCI51800.2020.00174

[11] A. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, pp. 9–15, 2021. [Online]. Available: https://aclanthology.org/2021.osact-1.2

[12] M. A. Al-Badrashiny, A. R. Sadat, and N. Diab, "CAMeL Tools: An open-source toolkit for Arabic natural language processing," Natural Language Engineering, vol. 27, no. 4, pp. 589–608, 2021. [Online]. Available: https://doi.org/10.1017/S1351324921000189

[13] T. El-Halees, "Arabic text classification using machine learning and deep learning approaches," IEEE Access, vol. 8, pp. 158 420–158 429, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3019467

[14] A. Alharbi and A. Azmi, "A survey of Arabic text classification: Challenges and solutions," Information Processing & Management, vol. 57, no. 6, p. 102347, 2020. [Online]. Available: https://doi.org/10.1016/j.ipm.2020.102347

[15] M. S. Abuarqoub, M. Al-Ayyoub, and Y. Jararweh, "Deep learning approaches for Arabic sentiment analysis," Future Generation Computer Systems, vol. 118, pp. 344–353, 2021. [Online]. Available: https://doi.org/10.1016/j.future.2020.12.020